

Calculadora d'expressions aritmetiques

Generado por Doxygen 1.8.12

Índice general

1	Pràctica Programació 2: Calculadora d'expressions aritmètiques	1
2	Índice de clases	3
2.1	Lista de clases	3
3	Indice de archivos	5
3.1	Lista de archivos	5
4	Documentación de las clases	7
4.1	Referencia de la Clase operacio	7
4.1.1	Descripción detallada	7
4.1.2	Documentación del constructor y destructor	7
4.1.2.1	operacio()	7
4.1.3	Documentación de las funciones miembro	8
4.1.3.1	guardar_operacio()	8
4.1.3.2	operar()	8
4.2	Referencia de la Clase resultat	8
4.2.1	Descripción detallada	9
4.2.2	Documentación del constructor y destructor	9
4.2.2.1	resultat()	9
4.2.3	Documentación de las funciones miembro	9
4.2.3.1	es_bool()	9
4.2.3.2	es_llista()	10
4.2.3.3	es_enter()	10
4.2.3.4	es_indefinit()	10

4.2.3.5	c_resultat()	11
4.2.3.6	print()	11
4.3	Referencia de la Clase resultat_lectura	11
4.3.1	Descripción detallada	12
4.3.2	Documentación del constructor y destructor	12
4.3.2.1	resultat_lectura()	12
4.3.3	Documentación de las funciones miembro	12
4.3.3.1	es_def_var()	12
4.3.3.2	es_def_op()	12
4.3.3.3	es_expressio()	13
4.3.3.4	entrada()	13
4.3.3.5	fi_entrada()	13
4.4	Referencia de la Clase variable	14
4.4.1	Descripción detallada	14
4.4.2	Documentación del constructor y destructor	14
4.4.2.1	variable()	14
4.4.3	Documentación de las funciones miembro	14
4.4.3.1	guardar_variable()	14
4.4.3.2	consultar_var()	15
4.4.3.3	buscar()	15
5	Documentación de archivos	17
5.1	Referencia del Archivo main.cc	17
5.1.1	Descripción detallada	17
5.1.2	Documentación de las funciones	18
5.1.2.1	avaluar()	18
5.1.2.2	main()	18
5.2	Referencia del Archivo operacio.hh	19
5.3	Referencia del Archivo resultat.hh	19
5.4	Referencia del Archivo resultat_lectura.hh	20
5.5	Referencia del Archivo variable.hh	20
Índice		21

Capítulo 1

Pràctica Programació 2: Calculadora d'expressions aritmètiques

En aquesta pràctica hem implementat una calculadora d'expressions aritmètiques que, a més a més, permet definir noves operacions a partir d'operacions ja existents.

Hem fet servir les classes [operacio.hh](#), [variable.hh](#), [resultat_lectura.hh](#), [resultat.hh](#).

Capítulo 2

Índice de clases

2.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

operacio	Conté les operacions que s'han definit i les opera quan apareixen a una expressió	7
resultat	Objecte que representa el resultat d'una operació	8
resultat_lectura	Guarda la informació necessària per entendre i tractar l'entrada de la pràctica	11
variable	Conté les variables que s'han definit	14

Capítulo 3

Indice de archivos

3.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

main.cc	Programa principal para la practica <i>Calculadora d'expressions aritmetiques</i>	17
operacio.hh	19
resultat.hh	19
resultat_lectura.hh	20
variable.hh	20

Capítulo 4

Documentación de las clases

4.1. Referencia de la Clase operacio

Conté les operacions que s'han definit i les opera quan apareixen a una expressió

Métodos públicos

- `operacio ()`
Constructora per defecte.
- `bool guardar_operacio (const list< string > &expressio)`
Serveix per guardar una nova operació
- `resultat operar (list< string > &op, list< string >::iterator it)`
Opera una expressio.

4.1.1. Descripción detallada

Conté les operacions que s'han definit i les opera quan apareixen a una expressió

Definición en la línea 12 del archivo operacio.hh.

4.1.2. Documentación del constructor y destructor

4.1.2.1. operacio()

```
operacio::operacio ( )
```

Constructora per defecte.

Precondición

Cert

Postcondición

el p.i està buit.

4.1.3. Documentación de las funciones miembro

4.1.3.1. guardar_operacio()

```
bool operacio::guardar_operacio (
    const list< string > & expressio )
```

Serveix per guardar una nova operació

Precondición

expressio és una expressió vàlida tal i com s'explicita a l'enunciat de la pràctica.

Postcondición

true si ha aconseguit guardar l'operacio *expressio* amb el nom corresponent i treu per pantalla la sortida que indica el enunciat, altrament retorna *false*

4.1.3.2. operar()

```
resultat operacio::operar (
    list< string > & op,
    list< string >::iterator it )
```

Opera una expressio.

Precondición

it = IT, cert

Postcondición

Retorna el resultat de l'operació que comença a on apunta IT

La documentación para esta clase fue generada a partir del siguiente fichero:

- [operacio.hh](#)

4.2. Referencia de la Clase resultat

Objecte que representa el resultat d'una operació

Métodos públicos

- `resultat ()`
Constructora per defecte.
- `bool es_bool () const`
Consultora 1.
- `bool es_llista () const`
Consultora 2.
- `bool es_enter () const`
Consultora 3.
- `bool es_indefinit () const`
Consultora 4.
- `list< int > c_resultat () const`
Consultora 5.
- `void print () const`
treu per pantalla la informacio del resultat tal i com explicita l'enunciat

4.2.1. Descripción detallada

Objecte que representa el resultat d'una operació

Definición en la línea 11 del archivo resultat.hh.

4.2.2. Documentación del constructor y destructor

4.2.2.1. resultat()

```
resultat::resultat ( )
```

Constructora per defecte.

Precondición

Cert

Postcondición

el p.i és indefinit.

4.2.3. Documentación de las funciones miembro

4.2.3.1. es_bool()

```
bool resultat::es_bool ( ) const
```

Consultora 1.

Precondición

Cert

Postcondición

Retorna *true* si el p.i. representa un bool definit i *false* si no.

4.2.3.2. es_llista()

```
bool resultat::es_llista ( ) const
```

Consultora 2.

Precondición

Cert

Postcondición

Retorna *true* si el p.i. representa una llista definida i *false* si no.

4.2.3.3. es_enter()

```
bool resultat::es_enter ( ) const
```

Consultora 3.

Precondición

Cert

Postcondición

Retorna *true* si el p.i. representa un enter definit i *false* si no.

4.2.3.4. es_indefinit()

```
bool resultat::es_indefinit ( ) const
```

Consultora 4.

Precondición

Cert

Postcondición

Retorna *true* si el p.i. es indefinit i *false* si no.

4.2.3.5. c_resultat()

```
list<int> resultat::c_resultat ( ) const
```

Consultora 5.

Precondición

El p.i. no és indefinit

Postcondición

Retorna el resultat

4.2.3.6. print()

```
void resultat::print ( ) const
```

treu per pantalla la informacio del resultat tal i com explicita l'enunciat

Precondición

Cert

Postcondición

En el canal estandar de sortida es troba la informacio del resultat tal i com explicita l'enunciat

La documentación para esta clase fue generada a partir del siguiente fichero:

- [resultat.hh](#)

4.3. Referencia de la Clase resultat_lectura

Guarda la informació necessària per entendre i tractar l'entrada de la pràctica.

Métodos públicos

- [resultat_lectura](#) ()
Constructora. L'objecte creat conté l'entrada i informació sobre aquesta.
- bool [es_def_var](#) () const
Consultora 1.
- bool [es_def_op](#) () const
Consultora 2.
- bool [es_expressio](#) () const
Consultora 3.
- list< string > [entrada](#) () const
Consultora 4.
- bool [fi_entrada](#) () const
Consultora 5.

4.3.1. Descripción detallada

Guarda la informació necessària per entendre i tractar l'entrada de la pràctica.

Definición en la línea 11 del archivo `resultat_lectura.hh`.

4.3.2. Documentación del constructor y destructor

4.3.2.1. `resultat_lectura()`

```
resultat_lectura::resultat_lectura ( )
```

Constructora. L'objecte creat conté l'entrada i informació sobre aquesta.

Precondición

Al canal estàndard d'entrada hi ha una expressió o definició tal i com especifica l'enunciat

Postcondición

L'objecte `resultat_lectura` creat conté l'entrada i informació sobre quin tipus d'entrada és: una expressió, una definició d'operació o una definició de variable.

4.3.3. Documentación de las funciones miembro

4.3.3.1. `es_def_var()`

```
bool resultat_lectura::es_def_var ( ) const
```

Consultora 1.

Precondición

Cert

Postcondición

Retorna *true* si el p.i. representa una entrada que defineix una variable i *false* si no.

4.3.3.2. `es_def_op()`

```
bool resultat_lectura::es_def_op ( ) const
```

Consultora 2.

Precondición

Cert

Postcondición

Retorna *true* si el p.i. representa una entrada que defineix una operació i *false* si no.

4.3.3.3. es_expressio()

```
bool resultat_lectura::es_expressio ( ) const
```

Consultora 3.

Precondición

Cert

Postcondición

Retorna *true* si el p.i. representa una entrada que es una expressio i *false* si no.

4.3.3.4. entrada()

```
list<string> resultat_lectura::entrada ( ) const
```

Consultora 4.

Precondición

el p.i no esta buit

Postcondición

Retorna l'entrada en el format més adient segons el tipus.

4.3.3.5. fi_entrada()

```
bool resultat_lectura::fi_entrada ( ) const
```

Consultora 5.

Precondición

el p.i no esta buit

Postcondición

Retorna *true* si el p.i. representa una entrada que senyalitza el final de l'entrada i *false* si no.

La documentación para esta clase fue generada a partir del siguiente fichero:

- [resultat_lectura.hh](#)

4.4. Referencia de la Clase variable

Conté les variables que s'han definit.

Métodos públicos

- `variable ()`
Constructora per defecte.
- `bool guardar_variable (string nom, const resultat &r)`
Serveix per guardar una nova variable.
- `bool consultar_var (const list< string > &op, list< string >::iterator it) const`
Serveix per consultar si existeix una variable amb el nom nom
- `resultat buscar (const list< string > &op, list< string >::iterator it) const`
Retorna el resultat que representa una variable.

4.4.1. Descripción detallada

Conté les variables que s'han definit.

Definición en la línea 13 del archivo variable.hh.

4.4.2. Documentación del constructor y destructor

4.4.2.1. variable()

```
variable::variable ( )
```

Constructora per defecte.

Precondición

Cert

Postcondición

el p.i està buit.

4.4.3. Documentación de las funciones miembro

4.4.3.1. guardar_variable()

```
bool variable::guardar_variable (
    string nom,
    const resultat & r )
```

Serveix per guardar una nova variable.

Precondición

Cert

Postcondición

retorna *true* si ha aconseguit guardar el resultat *R* amb el nom *nom* i treu per pantalla la sortida que indica el enunciat, altrament retorna *false*

4.4.3.2. consultar_var()

```
bool variable::consultar_var (
    const list< string > & op,
    list< string >::iterator it ) const
```

Serveix per consultar si existeix una variable amb el nom *nom*

Precondición

cert

Postcondición

retorna *true* si existeix una variable amb el nom *nom*, altrament retorna *false*

4.4.3.3. buscar()

```
resultat variable::buscar (
    const list< string > & op,
    list< string >::iterator it ) const
```

Retorna el resultat que representa una variable.

Precondición

Cert,

Postcondición

Retorna el resultat que representa la variable apuntada per *it*

La documentació per esta clase fue generada a partir del siguiente fichero:

- [variable.hh](#)

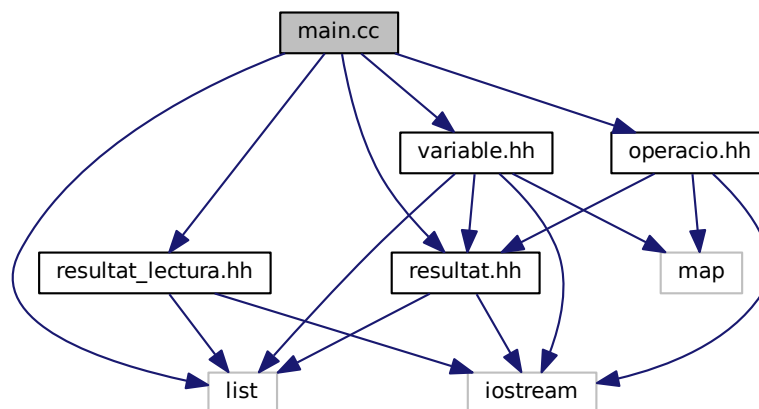
Capítulo 5

Documentación de archivos

5.1. Referencia del Archivo main.cc

Programa principal para la practica *Calculadora d'expressions aritmetiques*.

Dependencia gráfica adjunta para main.cc:



Funciones

- **resultat avaluar** (list< string > &op, list< string >::iterator &it)
Avalua una expressió
- int **main** ()
Programa principal de la pràctica.

5.1.1. Descripción detallada

Programa principal para la practica *Calculadora d'expressions aritmetiques*.

5.1.2. Documentación de las funciones

5.1.2.1. avaluar()

```
resultat avaluar (
    list< string > & op,
    list< string >::iterator & it )
```

Avalua una expressió

Precondición

it = IT

Postcondición

Retorna el resultat que representa l'expressió que comença a IT.

5.1.2.2. main()

```
int main ( )
```

Programa principal de la pràctica.

Precondición

L'entrada és tal i com s'indica a l'enunciat.

Postcondición

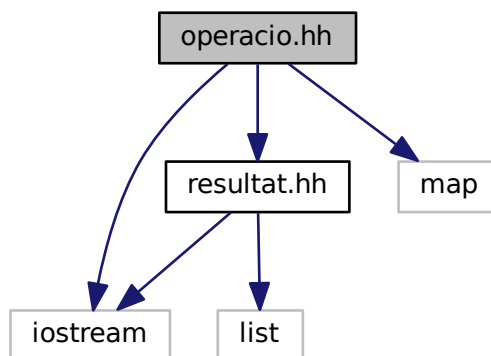
Al canal estàndard de sortida hi ha resoltes les diferents operacions i definicions.

Definición en la línea 33 del archivo main.cc.

```
33     {
34     operacio o;
35     variable v;
36     while(true){
37         resultat_lectura rl;
38         if(rl.fi_entrada()) return 0;
39         list<string> l = rl.entrada();
40         if(rl.es_def_var()){
41             resultat r;
42             list<string>::iterator it = l.begin();
43             string nom = *it;
44             ++it;
45             r = avaluar(l,it);
46             v.guardar_variable(nom , r);
47         }
48         else if(rl.es_def_op()){
49             o.guardar_operacio(l);
50         }
51         else if (rl.es_expressio()){
52             list<string>::iterator it = l.begin();
53             resultat r = avaluar(l, it);
54             r.print();
55         }
56     }
57 }
58
59 }
```

5.2. Referencia del Archivo operacio.hh

Dependencia gráfica adjunta para operacio.hh:



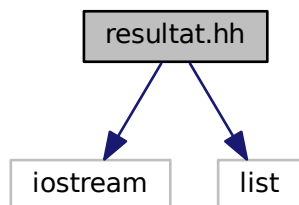
Clases

- class `operacio`

Conté les operacions que s'han definit i les opera quan apareixen a una expressió

5.3. Referencia del Archivo resultat.hh

Dependencia gráfica adjunta para resultat.hh:



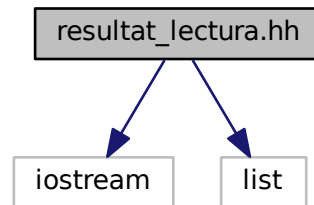
Clases

- class `resultat`

Objecte que representa el resultat d'una operació

5.4. Referencia del Archivo resultat_lectura.hh

Dependencia gráfica adjunta para resultat_lectura.hh:



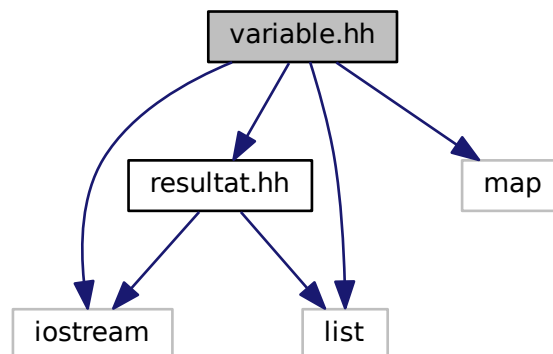
Clases

- class `resultat_lectura`

Guarda la informació necessària per entendre i tractar l'entrada de la pràctica.

5.5. Referencia del Archivo variable.hh

Dependencia gráfica adjunta para variable.hh:



Clases

- class `variable`

Conté les variables que s'han definit.

Índice alfabético

avaluar
 main.cc, 18

buscar
 variable, 15

c_resultat
 resultat, 10

consultar_var
 variable, 14

entrada
 resultat_lectura, 13

es_bool
 resultat, 9

es_def_op
 resultat_lectura, 12

es_def_var
 resultat_lectura, 12

es_enter
 resultat, 10

es_expressio
 resultat_lectura, 12

es_indefinit
 resultat, 10

es_llista
 resultat, 9

fi_entrada
 resultat_lectura, 13

guardar_operacio
 operacio, 8

guardar_variable
 variable, 14

main
 main.cc, 18

main.cc, 17
 avaluar, 18
 main, 18

operacio, 7
 guardar_operacio, 8
 operacio, 7
 operar, 8

operacio.hh, 19

operar
 operacio, 8

print
 resultat, 11

resultat, 8
 c_resultat, 10
 es_bool, 9
 es_enter, 10
 es_indefinit, 10
 es_llista, 9
 print, 11
 resultat, 9

resultat.hh, 19

resultat_lectura, 11
 entrada, 13
 es_def_op, 12
 es_def_var, 12
 es_expressio, 12
 fi_entrada, 13
 resultat_lectura, 12

resultat_lectura.hh, 20

variable, 14
 buscar, 15
 consultar_var, 14
 guardar_variable, 14
 variable, 14

variable.hh, 20